Appl. No. 09/540,952
Amdt. dated August 5, 2004
Reply to Office action of April 22, 2004

Listing of Claims:

1. (Currently amended) A method of monitoring the performance of a program being executed on a computer system, comprising:

executing the program on a computer system, the program having object code instructions;

at intervals interrupting execution of the program, including delivering a first interrupt; and

in response to at least a subset of the first interrupts, measuring a latency of execution of a particular object code instruction, storing the latency in a first database, the particular object code instruction being executed by the computer such that the program remains unmodified.

2. (Previously presented) The method of claim 1 wherein measuring the latency includes:

determining an initial value of a cycle counter;

performing the particular object code instruction;

determining a final value of the cycle counter; and

measuring the latency based on the initial value and the final value.

3. (Original) The method of claim 2 further comprising:

executing at least one instruction selected from the set consisting of (A) an instruction for ensuring that the particular object code instruction is performed after the initial value of the cycle counter is determined, and (B) an instruction for ensuring that the particular object code instruction is performed before the final value of the cycle counter is determined.

4. (Original) The method of claim 2 further comprising:

applying an adjustment to the final value.

5. (Cancelled).

6. (Original) The method of claim 1 wherein the particular object code instruction has a variable execution time.

7. (Original) The method of claim 1 wherein the particular object code instruction is a memory access instruction.

8. (Original) The method of claim 1 wherein the computer system includes a plurality of memory units, each memory unit of the plurality of memory units having a different range of access times, and further comprising:

associating the particular object code instruction with a memory unit in accordance with the latency and the range of access times for the memory unit.

9. (Previously presented) The method of claim 1 wherein measuring the latency includes:

determining an initial value of a cycle counter;

executing a first dependent instruction to provide a predetermined execution order;

performing the particular object code instruction;

executing a second dependent instruction to provide the predetermined execution order;

determining a final value of the cycle counter; and

determining the latency based on the initial value and the final value.

10. (Original) The method of claim 9 wherein the first and second dependent instructions are memory barrier instructions.

11. (Previously presented) The method of claim 1 wherein measuring includes:

identifying at least one issue block of instructions; and

interpreting the instructions of the at least one issue block;

wherein said particular object code instruction is in the issue block.

12. (Original) The method of claim 11 wherein said interpreting emulates a machine language instruction set of the computer system.

13. (Original) The method of claim 11 wherein said interpreting updates a state of the interrupted program as though each interpreted instruction had been directly executed by the computer system.

14. (Currently Amended) A computer program product for sampling latency of a computer program having object code instructions while the object code instructions are executing without modifying the computer program, the computer program product for use in conjunction with a computer system, the computer program product comprising a computer readable storage medium and a computer program mechanism embedded therein, the computer program mechanism comprising:

one or more instructions to deliver interrupts at intervals during execution of the program, including delivering a first interrupt;

one or more instructions to measure a latency of execution of a particular object code instruction; and

one or more instructions to, in response to at least a subset of the first interrupts, store the latency value for the particular object code instruction in a first database.

15. (Previously presented) The computer program product of claim 14 wherein said one or more instructions to measure the latency value include instructions to:

determine an initial value of a cycle counter;

perform the particular object code instruction;

determine a final value of the cycle counter; and

measure the latency based on the initial value and the final value.

16. (Original) The computer program product of claim 14 further comprising at least one instruction selected from the set consisting of (A) an instruction for ensuring that the particular object code instruction is performed after the initial value of the cycle counter is determined, and (B) an instruction for ensuring that the particular object code instruction is performed before the final value of the cycle counter is determined.

17. (Original) The computer program product of claim 15 further comprising one or more instructions to apply an adjustment to the final value.

18. (Cancelled).

19. (Original) The computer program product of claim 14 wherein the particular object code instruction has a variable execution time.

20. (Original) The computer program product of claim 14 wherein the particular object code instruction is a memory access instruction.

21. (Original) The computer program product of claim 14 wherein the computer system includes a plurality of memory units, each memory unit of the plurality of memory units having a different range of access times, and further comprising one or more instructions that associate the particular object code instruction with a memory unit in accordance with the latency value and the range of access times for the memory unit.

22. (Previously presented) The computer program product of claim 14 wherein said one or more instructions to measure the latency value include:

one or more instructions to determine an initial value of a cycle counter;
a first dependent instruction to provide a predetermined execution order;
the particular object code instruction;

a second dependent instruction to provide the predetermined execution order;

one or more instructions to determine a final value of the cycle counter; and

one or more instructions to measure the latency value based on the initial value and the final value.

23. (Original) The computer program product of claim 22 wherein the first and second dependent instructions are memory barrier instructions.

24. (Previously presented) The computer program product of claim 14 wherein said instructions to measure include:

one or more instructions that identify at least one issue block of instructions; and

an interpreter to interpret the instructions of the at least one issue block;

wherein said particular object code instruction is in the issue block.

25. (Original) The computer program product of claim 24 wherein the interpreter emulates a machine language instruction set of the computer system.

26. (Original) The computer program product of claim 24 wherein the interpreter updates a state of the interrupted program as though each interpreted instruction had been directly executed by the computer system.

27. (Currently Amended) A computer system comprising:

a processor for executing instructions; and

a memory storing instructions including:

one or more instructions to deliver interrupts at intervals during execution of the program, including delivering a first interrupt;

one or more instructions to measure a latency of execution of a particular object code instruction; and

one or more instructions to, in response to at least a subset of the first interrupts, store the latency value for the particular object code instruction in a first database.

28. (Previously presented) The computer system of claim 27 wherein said one or more instructions to measure the latency value include instructions to:

determine an initial value of a cycle counter;

perform the particular object code instruction;

determine a final value of the cycle counter; and

measure the latency based on the initial value and the final value.

29. (Original) The computer system of claim 27 wherein the memory further comprises at least one instruction selected from the set consisting of (A) an instruction for ensuring that the particular object code instruction is performed after the initial value of the cycle counter is determined, and (B) an instruction for ensuring that the particular object code instruction is performed before the final value of the cycle counter is determined.

30. (Original) The computer system of claim 28 wherein the memory further comprises one or more instructions to apply an adjustment to the final value.

31. (Cancelled).

32. (Original) The computer system of claim 27 wherein the particular object code instruction has a variable execution time.

33. (Original) The computer system of claim 27 wherein the particular object code instruction is a memory access instruction.

34. (Original) The computer system of claim 27 further comprising:

a plurality of memory units, each memory unit of the plurality of memory units having a different range of access times, and

wherein the memory further comprises one or more instructions that associate the particular object code instruction with a memory unit in accordance with the latency value and the range of access times for the memory unit.

35. (Previously presented) The computer system of claim 27 wherein said one or more instructions to measure the latency value include:

one or more instructions to determine an initial value of a cycle counter;

a first dependent instruction to provide a predetermined execution order;

the particular object code instruction;

a second dependent instruction to provide the predetermined execution order;

one or more instructions to determine a final value of the cycle counter; and

one or more instructions to measure the latency value based on the initial value and the final value.

36. (Original) The computer system of claim 36 wherein the first and second dependent instructions are memory barrier instructions.

37. (Previously presented) The computer system of claim 27 wherein said instructions to measure include:

one or more instructions that identify at least one issue block of instructions; and

an interpreter to interpret the instructions of the at least one issue block;

wherein said particular object code instruction is in the issue block.

38. (Original) The computer system of claim 37 wherein the interpreter emulates a machine language instruction set of the computer system.

39. (Original) The computer system of claim 37 wherein the interpreter updates a state of the interrupted program as though each interpreted instruction had been directly executed by the computer system.